

REMARKS

The applicants have carefully considered the Office action dated April 9, 2007. In the Office action, FIG. 1 was objected to, claims 1-15 were indicated to be allowable, and claims 16-34 were rejected under 35 U.S.C. § 101. In view of the following remarks, it is respectfully submitted that all pending claims are in condition for allowance and favorable reconsideration is respectfully requested.

Turning first to the objection to the drawings, the applicants respectfully submit that FIG. 1 is not limited to the illustration of the prior art. Rather, FIG. 1 is a block diagram of an example computer system that is programmed to implement the example methods and apparatus for reducing memory latency in a software application. A first processor system with particular programming is not the same a second processor system with different programming, even if the underlying hardware is the same. The specification describes the programming of the processor system of FIG. 1:

“FIG. 1 is a block diagram of an example computer system illustrating an environment of use for the disclosed system.” (Page 3, Para. [0010]).

“[A] software profiler and/or a software performance analysis may be executed by the computer system 100 to determine how well a software application is performing. The computer system 100 may also execute software applications that have been optimized for improved performance on the computer system 100.” (Page 5, Para. [0016]).

“Preferably, the illustrated process 300 is embodied in one or more software programs that are stored in one or more memories (e.g., flash memory 112 and/or hard disk 120) and executed by one or more processors (e.g., processor 106

and/or 202) in a well known manner.” (Page 7, Para. [0021]).

Accordingly, because FIG. 1 describes a computer system that includes more than what is “old,” the applicants respectfully request that the objection to the drawings be withdrawn.

The Office action rejects claims 16-34 under 35 U.S.C. § 101 as reciting non-statutory subject-matter. In particular, the Office action argues that claims 16-34 do not produce a tangible result because “the threads the inserted code is going into is not clearly and concisely claimed to be tangibly embodied on a computer readable medium.” The interim guidelines cited by the examiner do not state that the result of a claim must be “clearly and concisely claimed to be tangibly embodied on a compute readable medium” to be tangible. Rather, the guidelines state:

In determining whether the claim is for a “practical application,” the focus is not on whether the steps taken to achieve a particular result are useful, tangible and concrete, but rather that the final result achieved by the claimed invention is “useful, tangible and concrete.”

(Page 20, lines 8-11). The interim guidelines additionally state:

The tangible requirement does not necessarily mean that a claim must either be tied to a particular machine or apparatus or must operate to change articles or materials to a different state or thing. However, the tangible requirement does require that the claim must recite more than a § 101 judicial exception, in that the process claim must set forth a practical application of that § 101 judicial exception to produce a real-world result. Benson, 409 U.S. at 71-72, 175 USPQ at 676-77 (invention ineligible because had “no substantial practical application.”). “[A]n application of a law of nature or mathematical formula to a ... process may well be deserving of patent protection.” Diehr, 450 U.S. at 187, 209 USPQ at 8 (emphasis added); see also 21 Corning, 56 U.S. (15 How.) at

268, 14 L.Ed. 683 (“It is for the discovery or invention of some practical method or means of producing a beneficial result or effect, that a patent is granted . . .”). In other words, the opposite meaning of “tangible” is “abstract.”

(Page 21, subsection (2)) (emphasis added).

Claim 16 recites a system to reduce memory latency, the system comprising: a processor; a memory operatively coupled to the processor: the memory storing a software tool structured to identify a code region in an application program that suffers from a data cache miss; a compiler operatively coupled to the software tool, the compiler being structured to receive information from the software tool and to generate a helper thread; a set of compiler-runtime instructions to be generated and inserted in the application program to manage the helper thread and to manage a main thread; and a counting mechanism for insertion in the main thread and the helper thread to facilitate coordination of execution points associated with the helper thread and the main thread. The specification of the instant application states: “Generally, the disclosed system uses helper threads to prefetch variables and/or instructions within a main thread of a software application in order to reduce memory latency.” (Page 2, Para. [0009]).

In light of the forgoing citations, it is clear that claim 16 produces a real-world result. For example, one particular real-world result is the reduction of memory latency that is produced by the use of helper threads. The reduction of memory latency is not abstract because it has real-world implications on the execution of a software application.

In addition, the applicants note that, contrary to the statement in the Office action, claim 16 is not directed to software. Rather, claim 16 is

directed to a processor system including a memory storing a software tool, a compiler, a set of compiler-runtime instructions, and a counting mechanism. The software tool is clearly embodied in a computer readable memory that is operatively coupled to a processor. Annex IV of the interim guidelines states: “In this context, “functional descriptive material” consists of data structures and computer programs which impart functionality when employed as a computer component.” (Page 50, lines 3-5). Further,

When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory).

(Page 50, lines 13-22) (emphasis added). Claim 16 recites functional descriptive material (e.g., the software tool) that is recorded on a computer-readable medium (e.g., the memory operatively coupled to the processor). As previously described, the recitations of claim 16 provide at least the functionality of reducing memory latency in a processor system. Accordingly, claim 16 recites statutory functional descriptive material on computer readable medium.

In light of the forgoing remarks, claim 16 and all claims depending therefrom are statutory and in condition for allowance.

Claim 23 recites a machine readable medium storing instructions to cause a machine to: analyze a software application including a main thread; identify a code region in the software application; generate a helper thread; generate and insert a first set of compiler-runtime instructions in the main thread to manage the helper thread and the main thread; generate and insert a second set of compiler-runtime instructions in the helper thread to manage the helper thread and the main thread; and manage execution points of the helper thread and the main thread. For at least the reasons provided in conjunction with claim 16, claim 23 and all claims depending therefrom are statutory and in condition for allowance.

Claim 29 recites an apparatus to reduce memory latency, the apparatus comprising: a software tool structured to identify a code region in an application program that suffers from a data cache miss; a compiler operatively coupled to the software tool, the compiler being structured to receive information from the software tool and to generate a helper thread; a set of compiler-runtime instructions to be generated and inserted in the application program to manage the helper thread and to manage a main thread; and a counting mechanism for insertion in the main thread and the helper thread to facilitate coordination of execution points associated with the helper thread and the main thread. For at least the reasons provided in conjunction with claim 23, claim 29 and all claims depending therefrom are statutory and in condition for allowance.

If the Examiner is of the opinion that a telephone conference would expedite the prosecution of this case, the Examiner is invited to contact the undersigned at the number identified below.

Respectfully submitted,

HANLEY, FLIGHT & ZIMMERMAN, LLC.
Suite 2100
150 South Wacker Drive
Chicago, Illinois 60606
(312) 580-1020

By: /Michael W. Zimmerman/
Michael W. Zimmerman
Registration No. 57,993

July 9, 2007